09-15-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship .... Rui et al.
Applicant .... Microsoft Corporation
Attorney's Docket No. MS1-416US
Title: Annotating Programs for Automatic Summary Generation

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks,
Washington, D.C. 20231

From: Allan T. Sponseller (Tel. 509-324-9256; Fax 509-323-8979)
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Specification–title page, plus 48 pages, including 65 claims and Abstract
2. Transmittal letter including Certificate of Express Mailing
3. 7 Sheets Formal Drawings (Figs. 1-7)
4. Return Post Card

Large Entity Status [x]   Small Entity Status [ ]

Date: Sept. 13, 2000   By: [signature]
Allan T. Sponseller
Reg. No. 38,318

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL624352652

Date: 9-13-00   By: Lori A. Vierra
Lori A. Vierra

EL624352652

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Annotating Programs for Automatic Summary Generation

Inventor(s):

Yong Rui

Anoop Gupta

Alejandro Acero

ATTORNEY'S DOCKET NO. MS1-416US

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 60/153,730, filed September 13, 1999, entitled "MPEG-7 Enhanced Multimedia Access" to Yong Rui, Jonathan Grudin, Anoop Gupta, and Liwei He, which is hereby incorporated by reference.

TECHNICAL FIELD

This invention relates to audio/video programming and rendering thereof, and more particularly to annotating programs for automatic summary generation.

BACKGROUND OF THE INVENTION

Watching television has become a common activity for many people, allowing people to receive important information (e.g., news broadcasts, weather forecasts, etc.) as well as simply be entertained. While the quality of televisions on which programs are rendered has improved, so too have a wide variety of devices been developed and made commercially available that further enhance the television viewing experience. Examples of such devices include Internet appliances that allow viewers to "surf" the Internet while watching a television program, recording devices (either analog or digital) that allow a program to be recorded and viewed at a later time, etc.

Despite these advances and various devices, mechanisms for watching television programs are still limited to two general categories: (1) watching the program "live" as it is broadcast, or (2) recording the program for later viewing. Each of these mechanisms, however, limits viewers to watching their programs in the same manner as they were was broadcast (although possibly time-delayed).

Often times, however, people do not have sufficient time to watch the entirety of a recorded television program. By way of example, a sporting event such as a baseball game may take 2 or 2½ hours, but a viewer may only have ½ hour that he or she can spend watching the recorded game. Currently, the only way for the viewer to watch such a game is for the viewer to randomly select portions of the game to watch (e.g., using fast forward and/or rewind buttons), or alternatively use a "fast forward" option to play the video portion of the recorded game back at a higher speed than that at which it was recorded (although no audio can be heard). Such solutions, however, have significant drawbacks because it is extremely difficult for the viewer to know or identify which portions of the game are the most important for him or her to watch. For example, the baseball game may have only a handful of portions that are exciting, with the rest being uninteresting and not exciting.

The invention described below addresses these disadvantages, providing for annotating of programs for automatic summary generation.

SUMMARY OF THE INVENTION

Annotating programs for automatic summary generation is described herein.

In accordance with one aspect, audio/video programming content is made available to a receiver from a content provider, and meta data is made available to the receiver from a meta data provider. The content provider and meta data provider may be the same or different devices. The meta data corresponds to the programming content, and identifies, for each of multiple portions of the programming content, an indicator of a likelihood that the portion is an exciting

portion of the content. The meta data can be used, for example, to allow summaries of the programming content to be generated by selecting the portions having the highest likelihoods of being exciting portions.

According to another aspect, exciting portions of a sporting event are automatically identified based on sports-specific events and sports-generic events. The audio data of the sporting event is analyzed to identify sports-specific events (such as baseball hits if the sporting event is a baseball program) as well as sports-generic events (such as excited speech from an announcer). These sports-specific and sports-generic events are used together to identify the exciting portions of the sporting event.

According to another aspect, exciting segments of a baseball program are automatically identified. Various features are extracted from the audio data of the baseball program and selected features are input to an excited speech classification subsystem and a baseball hit detection subsystem. The excited speech classification subsystem identifies probabilities that segments of the audio data contain excited speech (e.g., from an announcer). The baseball hit detection subsystem identifies probabilities that multiple-frame groupings of the audio data include baseball hits. These two sets of probabilities are input to a probabilistic fusion subsystem that determines, based on both probabilities, a likelihood that each of the segments is an exciting portion of the baseball program. These probabilities can then be used, for example, to generate a summary of the baseball program.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 shows a programming distribution and viewing system in accordance with one embodiment of the invention;

Fig. 2 illustrates an example of a suitable operating environment in which the invention may be implemented;

Fig. 3 illustrates an exemplary programming content delivery architecture in accordance with certain embodiments of the invention;

Fig. 4 illustrates an exemplary automatic summary generation process in accordance with certain embodiments of the invention;

Fig. 5 illustrates part of an exemplary audio clip and portions from which features are extracted;

Fig. 6 illustrates exemplary baseball hit templates that may be used in accordance with certain embodiments of the invention; and

Fig. 7 is a flowchart illustrating an exemplary process for rendering a program summary to a user in accordance with certain embodiments of the invention.

DETAILED DESCRIPTION

General System

Fig. 1 shows a programming distribution and viewing system 100 in accordance with one embodiment of the invention. System 100 includes a video and audio rendering system 102 having a display device including a viewing area

104. Video and audio rendering system 102 represents any of a wide variety of devices for playing video and audio content, such as a traditional television receiver, a personal computer, etc. Receiver 106 is connected to receive and render content from multiple different programming sources. Although illustrated as separate components, rendering system 102 may be combined with receiver 106 into a single component (e.g., a personal computer or television). Receiver 106 may also be capable of storing content locally, in either analog or digital format (e.g., on magnetic tapes, a hard disk drive, optical disks, etc.).

While audio and video have traditionally been transmitted using analog formats over the airwaves, current and proposed technology allows multimedia content transmission over a wider range of network types, including digital formats over the airwaves, different types of cable and satellite systems (employing both analog and digital transmission formats), wired or wireless networks such as the Internet, etc.

Fig. 1 shows several different physical sources of programming, including a terrestrial television broadcasting system 108 which can broadcast analog or digital signals that are received by antenna 110; a satellite broadcasting system 112 which can transmit analog or digital signals that are received by satellite dish 114; a cable signal transmitter 116 which can transmit analog or digital signals that are received via cable 118; and an Internet provider 120 which can transmit digital signals that are received by modem 122 via the Internet (and/or other network) 124. Both analog and digital signals can include programming made up of audio, video, and/or other data. Additionally, a program may have different components received from different programming sources, such as audio and video data from cable transmitter 116 but data from Internet provider 120. Other programming

sources might be used in different situations, including interactive television systems.

As described in more detail below, programming content made available to system 102 includes audio and video programs as well as meta data corresponding to the programs. The meta data is used to identify portions of the program that are believed to be exciting portions, as well as how exciting these portions are believed to be relative to one another. The meta data can be used to generate summaries for the programs, allowing the user to view only the portions of the program that are determined to be the most exciting.

Exemplary Operating Environment

Fig. 2 illustrates an example of a suitable operating environment in which the invention may be implemented. The illustrated operating environment is only one example of a suitable operating environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Other well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, programmable consumer electronics (e.g., digital video recorders), gaming consoles, cellular telephones, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

Alternatively, the invention may be implemented in hardware or a combination of hardware, software, and/or firmware. For example, one or more

application specific integrated circuits (ASICs) could be designed or programmed to carry out the invention.

Fig. 2 shows a general example of a computer 142 that can be used in accordance with the invention. Computer 142 is shown as an example of a computer that can perform the functions of receiver 106 of Fig. 1, or of one of the programming sources of Fig. 1 (e.g., Internet provider 120). Computer 142 includes one or more processors or processing units 144, a system memory 146, and a bus 148 that couples various system components including the system memory 146 to processors 144.

The bus 148 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory 146 includes read only memory (ROM) 150 and random access memory (RAM) 152. A basic input/output system (BIOS) 154, containing the basic routines that help to transfer information between elements within computer 142, such as during start-up, is stored in ROM 150. Computer 142 further includes a hard disk drive 156 for reading from and writing to a hard disk, not shown, connected to bus 148 via a hard disk drive interface 157 (e.g., a SCSI, ATA, or other type of interface); a magnetic disk drive 158 for reading from and writing to a removable magnetic disk 160, connected to bus 148 via a magnetic disk drive interface 161; and an optical disk drive 162 for reading from and/or writing to a removable optical disk 164 such as a CD ROM, DVD, or other optical media, connected to bus 148 via an optical drive interface 165. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data

for computer 142. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 160 and a removable optical disk 164, it will be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 160, optical disk 164, ROM 150, or RAM 152, including an operating system 170, one or more application programs 172, other program modules 174, and program data 176. A user may enter commands and information into computer 142 through input devices such as keyboard 178 and pointing device 180. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 144 through an interface 168 that is coupled to the system bus (e.g., a serial port interface, a parallel port interface, a universal serial bus (USB) interface, etc.). A monitor 184 or other type of display device is also connected to the system bus 148 via an interface, such as a video adapter 186. In addition to the monitor, personal computers typically include other peripheral output devices (not shown) such as speakers and printers.

Computer 142 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 188. The remote computer 188 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 142, although

only a memory storage device 190 has been illustrated in Fig. 2. The logical connections depicted in Fig. 2 include a local area network (LAN) 192 and a wide area network (WAN) 194. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. In certain embodiments of the invention, computer 142 executes an Internet Web browser program (which may optionally be integrated into the operating system 170) such as the "Internet Explorer" Web browser manufactured and distributed by Microsoft Corporation of Redmond, Washington.

When used in a LAN networking environment, computer 142 is connected to the local network 192 through a network interface or adapter 196. When used in a WAN networking environment, computer 142 typically includes a modem 198 or other means for establishing communications over the wide area network 194, such as the Internet. The modem 198, which may be internal or external, is connected to the system bus 148 via a serial port interface 168. In a networked environment, program modules depicted relative to the personal computer 142, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Computer 142 also includes a broadcast tuner 200. Broadcast tuner 200 receives broadcast signals either directly (e.g., analog or digital cable transmissions fed directly into tuner 200) or via a reception device (e.g., via antenna 110 or satellite dish 114 of Fig. 1).

Computer 142 typically includes at least some form of computer readable media. Computer readable media can be any available media that can be accessed by computer 142. By way of example, and not limitation, computer readable

media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other media which can be used to store the desired information and which can be accessed by computer 142. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The invention has been described in part in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the

program modules may be combined or distributed as desired in various embodiments.

For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

Content Delivery Architecture

Fig. 3 illustrates an exemplary programming content delivery architecture in accordance with certain embodiments of the invention. A client 220 receives programming content including both audio/video data 222 and meta data 224 that corresponds to the audio/video data 222. In the illustrated example, an audio/video data provider 226 is the source of audio/video data 222 and a meta data provider 228 is the source of meta data 224. Alternatively, meta data 224 and audio/video data 222 may be provided by the same source, or alternatively three or more different sources.

The data 222 and 224 can be made available by providers 226 and 228 in any of a wide variety of formats. In one implementation, data 222 and 224 are formatted in accordance with the MPEG-7 (Moving Pictures Expert Group) format. The MPEG-7 format standardizes a set of Descriptors (Ds) that can be used to describe various types of multimedia content, as well as a set of Description Schemes (DSs) to specify the structure of the Ds and their relationship. In MPEG-7, the audio and video data 222 are each described as one or more Descriptors, and the meta data 224 is described as a Description Scheme.

Client 220 includes one or more processor(s) 230 and renderer(s) 232. Processor 230 receives audio/video data 222 and meta data 224 and performs any necessary processing on the data prior to providing the data to renderer(s) 232. Each renderer 232 renders the data it receives in a human-perceptive manner (e.g., playing audio data, displaying video data, etc.). The processing of data 222 and 224 can vary, and can include, for example, separating the data for delivery to different renderers (e.g., audio data to a speaker and video data to a display device), determining which portions of the program are most exciting based on the meta data (e.g., probabilities included as the meta data), selecting the most exciting segments based on a user-desired summary presentation time (e.g., the user wants a 20-minute summary), etc.

Client 220 is illustrated as separate from providers 226 and 228. This separation can be small (e.g., across a LAN) or large (e.g., a remote server located in another city or state). Alternatively, data 222 and/or 224 may be stored locally by client 220, either on another device such as an analog or digital video recorder (not shown) coupled to client 220 or within client 220 (e.g., on a hard disk drive).

A wide variety of meta data 224 can be associated with a program. In the discussions below, meta data 224 is described as being "excited segment probabilities" which identify particular segments of the program and a corresponding probability or likelihood that each segment is an "exciting" segment. An exciting segment is a segment of the program believed to be typically considered exciting to viewers. By way of example, baseball hits are believed to be typically considered exciting segments of a baseball program.

The excited segment probabilities in meta data 224 can be generated in any of a variety of manners. In one implementation, the excited segment probabilities

are generated manually (e.g., by a producer or other individual(s) watching the program and identifying the exciting segments and assigning the corresponding probabilities). In another implementation, the excited segment probabilities are generated automatically by a process described in more detail below. Additionally, the excited segment probabilities can be generated after the fact (e.g., after a baseball game is over and its entirety is available on a recording medium), or alternatively on the fly (e.g., a baseball game may be monitored and probabilities generated as the game is played).

Automatic Summary Generation

The automatic summary generation process described below refers to sports-generic and sports-specific events, and refers specifically to the example of a baseball program. Alternatively, summaries can be automatically generated in an analogous manner for other programs, including other sporting events.

The automatic summary generation process analyzes the audio data of the baseball program and attempts to identify segments that include speech, and of those segments which can be identified as being "excited" speech (e.g., the excitement in an announcer's voice). Additionally, based on the audio data segments that include baseball hits are also identified. These excited speech segments and baseball hit segments are then used to determine, for each of the excited speech segments, a probability that the segment is truly an exciting segment of the program. Given these probabilities, a summary of the program can be generated.

Fig. 4 illustrates an exemplary automatic summary generation process in accordance with certain embodiments of the invention. The generation process

begins with the raw audio data 250 (also referred to as a raw audio clip), such as the audio portion of data 222 of Fig. 3. The raw audio data 250 is the audio portion of the program for which the summary is being automatically generated. The audio data 250 is input to feature extractor 252 which extracts various features from portions of audio data 250. In one implementation, feature extractor 252 extracts one or more of energy features, phoneme-level features, information complexity features, and prosodic features.

Fig. 5 illustrates part of an exemplary audio clip and portions from which features are extracted. Audio clip 258 is illustrated. Audio features are extracted from audio clip 258 using two different resolutions: a sports-specific event detection resolution used to assist in the identification of potentially exciting sports-specific events, and a sports-generic event detection resolution used to assist in the identification of potentially exciting sports-generic events. In the illustrated example, the sports-specific event detection resolution is 10 milliseconds (ms), while the sports-generic event detection resolution is 0.5 seconds. Alternatively, other resolutions could be used.

As used herein, the sports-specific event detection is based on 10 ms "frames", while the sports-generic event detection is based on 0.5 second "windows". As illustrated in Fig. 5, the 10 ms frames are non-overlapping and the 0.5 second windows are non-overlapping, although the frames overlap the windows (and vice versa). Alternatively, the frames may overlap other frames, and/or the windows may overlap other windows.

Returning to Fig. 4, feature extractor 252 extracts different features from audio data 250 based on both frames and windows of audio data 250. Exemplary features which can be extracted by feature extractor 252 are discussed below.

Different embodiments can use different combinations of these features, or alternatively use only selected ones of the features or additional features.

Extractor 252 extracts energy features for each of the 10ms frames of audio data 250, as well as for each of the 0.5 second windows. For each frame or window, feature vectors having, for example, one element are extracted that identify the short-time energy in each of multiple different frequency bands. The short-time energy for each frequency band is the average waveform amplitude in the frequency band over the given time period (e.g., 10ms frame or 0.5 second window). In one implementation, four different frequency bands are used: 0hz – 630hz, 630hz – 1720hz, 1720hz – 4400hz, and 4400hz and above, referred to as E_1, E_2, E_3, and E_4, respectively. An additional feature vector is also calculated as the summation of E_2 and E_3, referred to as E_{23}.

The energy features extracted for each of the 10ms frames are also used to determine energy statistics regarding each of the 0.5 second windows. Exemplary energy statistics extracted for each frequency band E_1, E_2, E_3, E_4, and E_{23} for the 0.5 second window are illustrated in Table I.

Table I

Statistic	Description
maximum energy	The highest energy value of the frames in the window.
average energy	The average energy value of the frames in the window.
energy dynamic range	The energy range over the frames in the window (the difference between the maximum energy value and a minimum energy value).

Extractor 252 extracts phoneme-level features for each of the 10ms frames of audio data 250. For each frame, two well-known feature vectors are extracted: a Mel-frequency Cepstral coefficient (MFCC) and the first derivative of the MFCC (referred to as the delta MFCC). The MFCC is the *cosine* transform of the pitch of the frame on the "Mel-scale", which is a gradually warped linear spectrum (with coarser resolution at high frequencies).

Extractor 252 extracts information complexity features for each of the 10 ms frames of audio data 250. For each frame, a feature vector representing the entropy (Etr) of the frame is extracted. For an N-point Fast Fourier Transform (FFT) of an audio signal $s(t)$, with $S(n)$ representing the nth frequency's component, entropy is defined as:

$$Etr = \sum_{n=1}^{N} P_n \log P_n$$

where:

$$P_n = \frac{|S(n)|^2}{\sum_{n=1}^{N} |S(n)|^2}$$

Extracting feature vectors representing entropy is well-known to those skilled in the art and thus will not be discussed further except as it relates to the present invention.

Extractor 252 extracts prosodic features for each of the 0.5 second windows of audio data 250. For each window, a feature vector representing the pitch (Pch) of the window is extracted. A variety of different well-known approaches can be

used in determining pitch, such as the auto-regressive model, the average magnitude difference function, the maximum *a posteriori* (MAP) approach, etc.

The pitch is also determined for each 10ms frame of the 0.5 second window. These individual frame pitches are then used to extract pitch statistics regarding the pitch of the window. Exemplary pitch statistics extracted for each 0.5 second window are illustrated in Table II.

Table II

Statistic	Description
non-zero pitch count	The number of frames in the window that have a non-zero pitch value.
maximum pitch	The highest pitch value of the frames in the window.
minimum pitch	The lowest pitch value of the frames in the window.
average pitch	The average pitch value of the frames in the window.
pitch dynamic range	The pitch range over the frames in the window (the difference between the maximum and minimum pitch values).

Selected ones of the extracted features are passed by feature extractor 252 to an excited speech classification subsystem 260 and a baseball hit detection subsystem 262. Excited speech classification subsystem 260 attempts to identify segments of the audio data that include excited speech (sports-generic events), while baseball hit detection subsystem 262 attempts to identify segments of the audio data that include baseball hits (sports-specific events). The segments identified by subsystems 260 and 262 may be of the same or alternatively different sizes (and may be varying sizes). Probabilities generated for the segments are then

input to a probabilistic fusion subsystem 264 to determine a probability that the segments are exciting.

Excited speech classification subsystem 260 uses a two-stage process to identify segments of excited speech. In a first stage, energy and phoneme-level features 266 from feature extractor 252 are input to a speech detector 268 that identifies windows of the audio data that include speech (speech windows 270). In the illustrated example, speech detector 268 uses both the E_{23} and the delta MFCC feature vectors. For each 0.5 second window, if the E_{23} and delta MFCC vectors each exceed corresponding thresholds, the window is identified as a speech window 270; otherwise, the window is classified as not including speech. In one implementation, the thresholds used by speech detector 268 are 2.0 for the delta MFCC feature, and 0.07**Ecap* for the E_{23} feature (where *Ecap* is the highest E_{23} value of all the frames in the audio clip (or alternatively all of the frames in the audio clip that have been analyzed so far), although different thresholds could alternatively be used.

In alternative embodiments, speech detector 268 may use different features to classify segments as speech or not speech. By way of example, energy only may be used (e.g., the window is classified as speech only if E_{23} exceeds a threshold amount (such as 0.2**Ecap*). By way of another example, energy and entropy features may both be used (e.g., the window is classified as speech only if the product of E_{23} and *Etr* exceeds a threshold amount (such as 50,000).

In the second stage, pitch and energy features 272, received from feature extractor 252, for each of the speech windows 270 are used by excited speech classifier 274 to determine a probability that each speech window 270 is excited speech. Classifier 274 then combines these probabilities to identify a probability

that a group of these windows (referred to as a segment, which in one implementation is five seconds) is excited speech. Classifier 274 outputs an indication of these excited speech segments 276, along with their corresponding probabilities, to probabilistic fusion subsystem 264.

Excited speech classifier 274 uses six statistics regarding the energy E_{23} features and the pitch (*Pch*) features extracted from each speech window 270: maximum energy, average energy, energy dynamic range, maximum pitch, average pitch, and pitch dynamic range. Classifier 274 concatenates these six statistics together to generate a feature vector (having nine elements or dimensions) and compares the feature vector to a set of training vectors (based on corresponding features of training sample data) in two different classes: an excited speech class and a non-excited speech class. The *posterior* probability of a feature vector X (for a window 270) being in a class C_i, where C_1 is the class of excited speech and C_2 is the class of non-excited speech, can be represented as: $P(C_i \mid X)$. The probability of error in classifying the feature vector X can be reduced by classifying the data to the class having the *posterior* probability that is the highest.

Speech classifier 274 determines the *posterior* probability $P(C_i \mid X)$ using learning machines. A wide variety of different learning machines can be used to determine the *posterior* probability $P(C_i \mid X)$. Three such learning machines are described below, although other learning machines could alternatively be used.

The *posterior* probability $P(C_i \mid X)$ can be determined using parametric machines, such as Bayes rule:

$$P(C_i \mid X) = \frac{P(C_i)p(X \mid C_i)}{p(X)}$$

where $p(X)$ is the data density, $P(C_i)$ is the prior probability, and $p(X \mid C_i)$ is the conditional class density. The data density $p(x)$ is a constant for all the classes and thus does not contribute to the decision rule. The prior probability $P(C_i)$ can be estimated from labeled training data (e.g., excited speech and non-excited speech) in a conventional manner. The conditional class density $p(X \mid C_i)$ can be calculated in a variety of different manners, such as the Gaussian (Normal) distribution $N(\mu,\sigma)$. The μ parameter (mean) and the σ parameter (standard deviation) can be determined using the well-known Maximum Likelihood Estimation (MLE):

$$\mu = \frac{1}{n}\sum_{k=1}^{n} X_k$$

$$\sigma^2 = \frac{1}{n}\sum_{k=1}^{n}(X_k - \mu)$$

where n is the number of training samples and X represents the training samples.

Another type of machines that can be used to determine the *posterior* probability $P(C_i \mid X)$ are non-parametric machines. The K nearest neighbor technique is an example of such a machine. Using the K nearest neighbor technique:

$$P(C_i \mid X) = \frac{\frac{K_i}{nV}}{\sum_i \frac{K_i}{nV}} = \frac{K_i}{K}$$

where V is the volume around feature vector X, V covers K labeled (training) samples, and K_i is the number of samples in class C_i.

Another type of machines that can be used to determine the *posterior* probability $P(C_i \mid X)$ are semi-parametric machines, which combine the advantages of non-parametric and parametric machines. Examples of such semi-parametric machines include Gaussian mixture models, neural networks, and support vector machines (SVMs).

Any of a wide variety of well-known training methods can be used to train the SVM. After the SVM is trained, a sigmoid function is trained to map the SVM outputs into *posterior* probabilities. The *posterior* probability $P(C_i \mid X)$ can then be determined as follows:

$$P(C_i \mid X) = \frac{1}{1 + \exp(AX + B)}$$

where A and B are the parameters of the sigmoid function. The parameters A and B are determined by reducing the negative log likelihood of training data (f_i, t_i), which is a cross-entropy error function:

$$\min - \sum_i t_i \log(p_i) + (1 - t_i)\log(1 - p_i)$$

where

$$p_i = \frac{1}{1 + \exp(Af_i + B)}$$

The cross-entropy error function minimization can be performed using any number of conventional optimization processes. The training data (f_i, t_i) can be the same training data used to train the SVM, or other data sets. For example, the

training data (f_i, t_i) can be a hold out set (in which a fraction of the initial training set, such as 30%, is not used to train the SVM but is used to train the sigmoid) or can be generated using three-fold cross-validation (in which the initial training set is split into three parts, each of three SVMs is trained on permutations of two out of three parts, and the f_i are evaluated on the remaining third, and the union of all three sets f_i forming the training set of the sigmoid).

Additionally, an out-of-sample model is used to avoid "overfitting" the sigmoid. Out-of-sample data is modeled with the same empirical density as the sigmoid training data, but with a finite probability of opposite label. In other words, when a positive example is observed at a value f_i, rather than using $t_i=1$, it is assumed that there is a finite chance of opposite label at the same f_i in the out-of-sample data. Therefore, a value of $t_i=1-\epsilon_+$ is used, for some ϵ_+. Similarly, a negative example will use a target value of $t_i=\epsilon_-$.

Regardless of the manner in which the *posterior* probability $P(C_i \mid X)$ for a 0.5 second window is determined, the *posterior* probabilities for multiple windows are combined to determine the *posterior* probability for a segment. In one implementation, each segment is five seconds, so the *posterior* probabilities of ten adjacent windows are used to determine the *posterior* probability for each segment.

The *posterior* probabilities for the multiple windows can be combined in a variety of different manners. In one implementation, the *posterior* probability of the segment being an exciting segment, referred to as *P(ES)*, is determined by averaging the *posterior* probabilities of the windows in the segment:

$$P(ES) = \frac{1}{M}\sum_{m=1}^{M} P(C_1 \mid X_m)$$

where C_1 represents the excited speech class and M is the number of windows in the segment.

Which ten adjacent windows to use for a segment can be determined in a wide variety of different manners. In one implementation, if ten or more adjacent windows include speech, then those adjacent windows are combined into a single segment (e.g., which may be greater than ten windows, or, if too large, which may be pared down into multiple smaller ten-window segments). However, if there are fewer than ten adjacent windows, then additional windows are added (before and/or after the adjacent windows, between multiple groups of adjacent windows, etc.) to get the full ten windows, with the *posterior* probability for each of these additional windows being zero.

The probabilities $P(ES)$ of these segments including excited speech 276 (as well as an indication of where these segments occur in the raw audio clip 250) are then made available to probabilistic fusion subsystem 264. Subsystem 264 combines the probabilities 276 with information received from baseball hit detection subsystem 262, as discussed in more detail below.

Baseball hit detection subsystem 262 uses energy features 278 from feature extractor 252 to identify baseball hits within the audio data 250. In one implementation, the energy features 278 include the E_{23} and E_4 features discussed above. Two additional features are also generated, which may be generated by feature extractor 252 or alternatively another component (not shown). These additional features are referred to as ER_{23} and ER_4, and are discussed in more detail below.

Hit detection is performed by subsystem 262 based on 25-frame groupings. A sliding selection of 25 consecutive 10ms frames of the audio data 250 is

analyzed, with the frame selection sliding frame-by-frame through the audio data 250. The features of the 25-frame groupings and a set of hit templates 280 are input to template matcher 282. Template matcher 282 compares the features of each 25-frame grouping to the hit templates 280, and based on this comparison determines a probability as to whether the particular 25-frame grouping contains a hit. An identification of the 25-frame groupings (e.g., the first frame in the grouping) and their corresponding probabilities are output by template matcher 282 as hit candidates 284.

Multiple-frame groupings are used to identify hits because the sound of a baseball hit is typically longer in duration than a single frame (which is, for example, only 10 ms). The baseball hit templates 280 are established to capture the shape of the energy curves (using the four energy features discussed above) over the time of the groupings (e.g., 25 10ms frames, or 0.25 seconds). Baseball hit templates 280 are designed so that the hit peak (the energy peak) is at the 8^{th} frame of the 25-frame grouping. The additional features ER_{23} and ER_4 are calculated by normalizing the E_{23} and E_4 features based on the energy features in the 8^{th} frame as follows:

$$ER_{23}(i) = \frac{E_{23}(i)}{E_{23}(8)}$$

$$ER_4(i) = \frac{E_4(i)}{E_4(8)}$$

where i ranges from 1 to 25, $E_{23}(8)$ is the E_{23} energy in the 8^{th} frame, and $E_4(8)$ is the E_4 energy in the 8^{th} frame.

Fig. 6 illustrates exemplary baseball hit templates 280 that may be used in accordance with certain embodiments of the invention. The templates 280 in Fig. 6 illustrate the shape of the energy curves over time (25 frames) for each of the four features E_{23}, E_4, ER_{23}, and ER_4.

For each group of frames, template matcher 282 determines the probability that the group contains a baseball hit. This can be accomplished in multiple different manners, such as un-directional or directional template mapping. Initially, the four feature vectors for each of the 25 frames are concatenated, resulting in a 100-element vector. The templates 280 are similarly concatenated for each of the 25 frames, also resulting in a 100-element vector. The probability of a baseball hit in a grouping $P(HT)$ can be calculated based on the Mahalanobis distance D between the concatenated feature vector and the concatenated template vector as follows:

$$D^2 = (\vec{X} - \vec{T})^T \Sigma^{-1} (\vec{X} - \vec{T})$$

where $\vec{X}$ is the concatenated feature vector, $\vec{T}$ is the concatenated template vector, and Σ is the covariance matrix of $\vec{T}$. Additionally, Σ is restricted to being a diagonal matrix, allowing the baseball hit probability $P(HT)$ to be determined as follows:

$$P(HT) = \frac{\exp(-\frac{1}{2} D^2)}{C + \exp(-\frac{1}{2} D^2)}$$

where C is a constant that is data dependent (e.g., exp(-0.5D'^2), where D'^2 is the distance between the concatenated feature vector and a template for non-hit signals).

Alternatively, a directional template matching approach can be used, with the distance D being calculated as follows:

$$D^2 = (\vec{X} - \vec{T})^T I \times \Sigma^{-1} (\vec{X} - \vec{T})$$

where I is a diagonal indicator matrix. The indicator matrix I is adjusted to account for over-mismatches or under-mismatches (an over-mismatch is actually good). In one implementation, when the values of E_{23} for the 25-frame grouping are overmatching the templates (e.g., more than a certain number (such as one-half) of the data values in the 25-frame grouping are higher than the corresponding template values), then $I = diag[1, \ldots, 1, -1, 1, \ldots, 1]$ where the –1 is at location 8. However, when the values of E_{23} for the 25-frame grouping are under-matching the templates (e.g., less than a certain number (such as one-half) of the data values in the 25-frame grouping are less than the corresponding template values), then $I = diag[-1, \ldots, -1, -1, -1, \ldots, -1]$ where the 1 is at location 8.

Although hit detection is described as being performed across all of the audio data 250, alternatively hit detection may be performed on only selected portions of the audio data 250. By way of example, hit detection may only be performed on the portions of audio data 250 that are excited speech segments (or speech windows) and for a period of time (e.g., five seconds) prior to those excited speech segments (or speech windows).

Probabilistic fusion generator 286 of subsystem 264 receives the excited speech segment probabilities *P(ES)* from excited speech classification subsystem 260 and the baseball hit probabilities *P(HT)* from baseball hit detection subsystem 262 and combines those probabilities to identify probabilities *P(E)* that segments of the audio data 250 are exciting. Probabilistic fusion generator 286 searches for hit frames within the 5-second interval of the excited speech segment. This combining is also referred to herein as "fusion".

Two different types of fusion can be used: weighted fusion and conditional fusion. Weighted fusion applies weights to each of the probabilities *P(ES)* and *P(HT)* adds the results to obtain the value *P(E)* as follows:

$$P(E) = W_{ES}P(ES) + W_{HT}P(HT)$$

where the weights W_{ES} and W_{HT} sum up to 1.0. In one implementation, W_{ES} is 0.83 and W_{HT} is 0.17, although other weights could alternatively be used.

Conditional fusion, on the other hand, accounts for the detected baseball hits adjusting the confidence level of the *P(ES)* estimation (e.g., that the excited speech probability is not high due to mislabeling a car horn as speech). The conditional fusion is calculated as follows:

$$\begin{aligned} &P(E) = P(CF)P(ES) \\ &P(CF) = P(CF \mid HT)P(HT) + P(CF \mid \overline{HT})P(\overline{HT}) \\ &P(\overline{HT}) = 1 - P(HT) \end{aligned}$$

where *P(CF)* is the probability of how much confidence there is in the *P(ES)* estimation, and $P(\overline{HT})$ is the probability that there is no hit. *P(CF|HT)* represents the probability that we are confident that *P(ES)* is accurate given there is a

baseball hit. Similarly, $P(CF|\overline{HT})$ represents the probability that we are confident that *P(ES)* is accurate given there is no baseball hit. Both conditional probabilities $P(CF|HT)$ and $P(CF|\overline{HT})$ can be estimated from the training data. In one implementation, the value of $P(CF|HT)$ is 1.0 and the value of $P(CF|\overline{HT})$ is 0.3.

The final probability *P(E)* that a segment is an exciting segment is then output by generator 286, identifying the exciting segments 288. These final probabilities, and an indication of the segments they correspond to, are stored as the meta data 224 of Fig. 3.

The actual portions of the program rendered for a user as the summary of the program are based on these exciting segments 288. Various modifications may be made, however, to make the rendering smoother. Examples of such modifications include: starting rendering of the exciting segment a period of time (e.g., three seconds) earlier than the hit (e.g., to render the pitching of the ball); merging together overlapping segments; merging together close-by (e.g., within ten seconds) segments; etc.

Once the probabilities that segments are exciting are identified, the user can choose to view a summary or highlights of the program. Which segments are to be delivered as the summary can be determined locally (e.g., on the user's client computer) or alternatively remotely (e.g., on a remote server).

Additionally, various "pre-generated" summaries may be generated and maintained by remote servers. For example, a remote server may identify which segments to deliver if a 15-minute summary is requested and which segments to deliver if a 30-minute summary is requested, and then store these identifications. By pre-generating such summaries, if a user requests a 15-minute summary, then

the pre-generated indications simply need to be accessed rather than determining, at the time of request, which segments to include in the summary.

Fig. 7 is a flowchart illustrating an exemplary process for rendering a program summary to a user in accordance with certain embodiments of the invention. The acts of Fig. 7 may be implemented in software, and may be carried out by a receiver 106 of Fig. 1 or alternatively a programming source of Fig. 1 (e.g., Internet provider 120).

Initially, the user request for a summary is received along with parameters for the summary (act 300). The parameters of the summary identify what level of summary the user desires, and can vary by implementation. By way of example, a user may indicate as the summary parameters that he or she wants to be presented with any segments that have a probability of 0.75 or higher of being exciting segments. By way of another example, a user may indicate as the summary parameters that he or she wants to be presented with a 20-minute summary of the program.

The meta data corresponding to the program (the exciting segment probabilities *P(E)*) is then accessed (act 302), and the appropriate exciting segments identified based on the summary parameters (act 304). Once the appropriate exciting segments are identified, they are rendered to the user (act 306). The manner in which the appropriate exciting segments are identified can vary, in part based on the nature of the summary parameters. If the summary parameters indicate that all segments with a *P(E)* of 0.75 or higher should be presented, then all segments with a *P(E)* of 0.75 or greater are identified. If the summary parameters indicate that a 20-minute summary should be generated, then the appropriate segments are identified by determining (based on the *P(E)* of the

segments and the lengths of the segments) the segments having the highest $P(E)$ that have a combined length less than 20 minutes.

Conclusion

Although the description above uses language that is specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the invention.

CLAIMS

1. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to perform acts including:

receiving the audio portion of a sporting event;

classifying a set of segments of the audio portion as excited speech;

classifying a set of frame groupings as including baseball hits;

combining the set of segments and the set of frame groupings to identify probabilities for each segment that the segment is an exciting segment; and

saving an indication of the set of segments and the corresponding probabilities as meta data corresponding to the sporting event.

2. One or more computer readable media as recited in claim 1, wherein each segment includes at least ten 0.5-second windows of the audio portion.

3. One or more computer readable media as recited in claim 1, wherein each frame grouping is a grouping of 25 10-millisecond frames of the audio portion.

4. One or more computer readable media as recited in claim 1, wherein the classifying a set of segments as excited speech comprises:

extracting a first set of features from the audio portion;

identifying, based on the first set of features, a plurality of windows of the audio portion that include speech;

extracting a second set of features from the audio portion; and

identifying, based on the second set of features, which of the plurality of windows include excited speech.

5. One or more computer readable media as recited in claim 4, wherein extracting the first set of features comprises, for each of a plurality of windows:

identifying an average waveform amplitude of the audio portion in a first frequency band of the window;

identifying an average waveform amplitude of the audio portion in a second frequency band of the window;

concatenating the identified average waveform amplitudes to generate an energy feature of the first set of features; and

determining, for an MFCC feature of the first set of features, the Mel-frequency Cepstral coefficient of the window.

6. One or more computer readable media as recited in claim 5, wherein the identifying a plurality of windows of the audio portion that include speech comprises determining that a window includes speech if the energy feature exceeds a first threshold and the MFCC feature exceeds a second threshold.

7. One or more computer readable media as recited in claim 4, wherein extracting the second set of features comprises, for each of a plurality of windows:

identifying, for each of a plurality of frames in the window, an average waveform amplitude of the audio portion in a first frequency band;

identifying, for each of the plurality of frames in the window, an average waveform amplitude of the audio portion in a second frequency band;

concatenating the identified average waveform amplitudes to generate an energy feature;

extracting, as a pitch feature of each frame, the pitch of each frame;

identifying a plurality of statistics regarding each window based on the energy features and pitch features of the plurality of frames.

8. One or more computer readable media as recited in claim 7, wherein the identifying a plurality of statistics further comprises:

identifying a maximum energy;

identifying an average energy;

identifying an energy dynamic range;

identifying a maximum pitch;

identifying an average pitch; and

identifying a dynamic pitch range.

9. One or more computer readable media as recited in claim 7, wherein the identifying which of the plurality of windows include excited speech comprises identifying a *posterior* probability that the window corresponds to an excited speech class and identifying a *posterior* probability that the window corresponds to a non-excited speech class, and classifying the window in the class with the higher *posterior* probability.

10. One or more computer readable media as recited in claim 4, further comprising instructions that cause the one or more processors to perform acts including outputting an excited speech probability for each of the segments that include excited speech, the excited speech probability for a segment indicating a likelihood that the segment includes excited speech.

11. One or more computer readable media as recited in claim 1, wherein the classifying a set of frame groupings as including baseball hits:

extracting, for each frame in a multiple-frame grouping, a set of features from the audio portion;

comparing the set of features from the multiple-frame groupings to a set of templates; and

identifying, for each of the multiple-frame groupings, a probability that the grouping includes a baseball hit based on how well the grouping matches the set of templates.

12. One or more computer readable media as recited in claim 11, wherein the extracting comprises, for each frame:

identifying an average waveform amplitude of the audio portion in a first frequency band of the frame;

identifying an average waveform amplitude of the audio portion in a second frequency band of the frame;

concatenating the identified average waveform amplitudes to generate a first energy feature of the set of features;

identifying an average waveform amplitude of the audio portion in a third frequency band of the frame; and

using, as a second energy feature of the set of features, the average waveform amplitude of the third frequency band.

13. One or more computer readable media as recited in claim 12, further comprising instructions that cause the one or more processors to perform acts including:

generating a third energy feature for each frame by normalizing the first energy feature based on the first energy feature of the eighth frame of the multiple-frame grouping; and

generating a fourth energy feature for each frame by normalizing the second energy feature based on the second energy feature of the eighth frame of the multiple-frame grouping.

14. One or more computer readable media as recited in claim 1, wherein the combining comprises generating, for each of the set of segments, a weighted sum of a probability that the segment includes excited speech and a probability that a frame grouping within the segment includes a baseball hit.

15. One or more computer readable media as recited in claim 1, wherein the combining comprises adjusting a probability that a segment of the set of segments includes excited speech based on a probability that the segment includes a baseball hit.

16. A method comprising:

receiving a program including both audio and video;

receiving meta data corresponding to the program; and

rendering, based on the meta data, portions of the program as a summary of the program.

17. A method as recited in claim 16, wherein the rendering comprises displaying the video of the portions and playing the audio of the portions.

18. A method as recited in claim 16, wherein the meta data comprises a probability indicator, for each of a plurality of portions of the program, that identifies a probability that the corresponding portion is an exciting portion of the program.

19. A method as recited in claim 18, wherein the rendering comprises selecting the plurality of portions that have probability indicators that exceed a threshold value, and rendering the selected portions as the summary.

20. A method as recited in claim 16, wherein the receiving a program and the receiving meta data comprise receiving both the program and the meta data from the same source.

21. A method as recited in claim 16, wherein the receiving meta data comprises receiving the meta data from a remote source via a network.

22. A method as recited in claim 21, wherein the network comprises the Internet.

23. A method as recited in claim 16, wherein the receiving meta data comprises receiving meta data generated manually.

24. A method as recited in claim 16, wherein the receiving meta data comprises receiving meta data generated automatically.

25. A method as recited in claim 16, wherein the meta data comprises a plurality of probabilities, each corresponding to a segment of the program, the probabilities representing a probabilistic combination of sports-specific events and sports-generic events identified in the program.

26. A method as recited in claim 25, wherein the sports-specific events comprise baseball hits, and wherein the sports-generic events comprise excited speech.

27. One or more computer readable media including a computer program that is executable by a processor to perform the method recited in claim 16.

28. A system comprising:

a content provider to make programming content available to requesting clients;

a meta data provider to make meta data, corresponding to the programming content, available to the requesting clients, wherein the meta data identifies, for each of a plurality of portions of the programming content, an indicator of a likelihood that the corresponding portion is an exciting portion of the programming content; and

a plurality of receivers coupled to receive the programming content from the content provider and the meta data from the meta data provider.

29. A system as recited in claim 28, wherein the content provider and the meta data provider are the same.

30. A system as recited in claim 28, wherein the content provider and the meta data provider are coupled to the plurality of receivers via the Internet.

31. A system as recited in claim 28, wherein the plurality of receivers are further to render, based on the meta data, portions of the programming content as a summary of the programming content.

32. A system as recited in claim 28, wherein the meta data comprises a probability indicator, for each of a plurality of portions of the programming content, that identifies a probability that the corresponding portion is an exciting portion of the program.

33. A system as recited in claim 32, wherein the plurality of receivers are further to select, from the plurality of portions, portions that have probability indicators that exceed a threshold value, and render the selected portions as the summary.

34. A system as recited in claim 28, wherein the meta data is generated manually.

35. A system as recited in claim 28, wherein the meta data is generated automatically.

36. A system as recited in claim 28, wherein the meta data comprises a plurality of probabilities, each corresponding to a segment of the programming content, the probabilities representing a probabilistic combination of sports-specific events and sports-generic events identified in the programming content.

37. A system as recited in claim 36, wherein the sports-specific events comprise baseball hits, and wherein the sports-generic events comprise excited speech.

38. A method of automatically summarizing a program, the method comprising:

identifying a plurality of sports-generic events from the audio of the program;

identifying a plurality of sports-specific events from the audio of the program; and

identifying, by combining the sports-generic events and the sports-specific events, a set of portions of the program as a summary of the program.

39. A method as recited in claim 38, further comprising transmitting the set of portions to a client computer as the summary of the program.

40. A method as recited in claim 39, wherein the transmitting comprises transmitting the set of portions via the Internet.

41. A method as recited in claim 38, wherein the sports-specific events comprise baseball hits, and wherein the sports-generic events comprise excited speech.

42. A method as recited in claim 38, wherein the program includes both an audio portion and a video portion.

43. One or more computer readable media including a computer program that is executable by a processor to perform the method recited in claim 38.

44. A method comprising:

analyzing audio data of a program to identify a first plurality of portions of the program each including excited speech;

analyzing the audio data to identify a second plurality of portions of the program each including a potential baseball hit; and

combining the first plurality of portions and the second plurality of portions to identify a set of segments of the program and a likelihood, for each of the segments in the set, that the segment is an exciting part of the program.

45. A method as recited in claim 44, wherein the analyzing audio data to identify the first plurality of portions comprises:

extracting a first set of features from the audio data;

identifying, based on the first set of features, a plurality of windows of the audio data that include speech;

extracting a second set of features from the audio data; and

identifying, based on the second set of features, which of the plurality of windows include excited speech.

46. A method as recited in claim 45, wherein extracting the first set of features comprises, for each of a plurality of windows:

identifying an average waveform amplitude of the audio data in a first frequency band of the window;

identifying an average waveform amplitude of the audio data in a second frequency band of the window;

concatenating the identified average waveform amplitudes to generate an energy feature of the first set of features; and

determining, for an MFCC feature of the first set of features, the Mel-frequency Cepstral coefficient of the window.

47. A method as recited in claim 46, wherein each window comprises 0.5 seconds.

48. A method as recited in claim 46, wherein the identifying a plurality of windows of the audio data that include speech comprises determining that a window includes speech if the energy feature exceeds a first threshold and the MFCC feature exceeds a second threshold.

49. A method as recited in claim 45, wherein extracting the second set of features comprises, for each of a plurality of windows:

identifying, for each of a plurality of frames in the window, an average waveform amplitude of the audio data in a first frequency band;

identifying, for each of the plurality of frames in the window, an average waveform amplitude of the audio data in a second frequency band;

concatenating the identified average waveform amplitudes to generate an energy feature;

extracting, as a pitch feature of each frame, the pitch of each frame; and

identifying a plurality of statistics regarding each window based on the energy features and pitch features of the plurality of frames.

50. A method as recited in claim 49, wherein the identifying a plurality of statistics further comprises:

identifying a maximum energy;

identifying an average energy;

identifying an energy dynamic range;

identifying a maximum pitch;

identifying an average pitch; and

identifying a dynamic pitch range.

51. A method as recited in claim 49, wherein the identifying which of the plurality of windows include excited speech comprises identifying a *posterior* probability that the window corresponds to an excited speech class and identifying a *posterior* probability that the window corresponds to a non-excited speech class, and classifying the window in the class with the higher *posterior* probability.

52. A method as recited in claim 51, wherein the *posteriori* probabilities are identified using a parametric machine.

53. A method as recited in claim 51, wherein the *posteriori* probabilities are identified using a non-parametric machine.

54. A method as recited in claim 51, wherein the *posteriori* probabilities are identified using a semi-parametric machine.

55. A method as recited in claim 45, further comprising outputting an excited speech probability corresponding to each of a plurality segments that include excited speech, each segment including a plurality of windows, the excited speech probability for a segment indicating a likelihood that the segment includes excited speech.

56. A method as recited in claim 55, wherein the excited speech probability is determined by averaging the *posterior* probabilities of each of the plurality of windows in the segment, the *posterior* probability of a window identifying the probability that the window includes excited speech.

57. A method as recited in claim 44, wherein the analyzing audio data to identify the second plurality of portions comprises:

extracting, for each frame in a multiple-frame grouping, a set of features from the audio data;

comparing the set of features from the multiple-frame groupings to a set of templates; and

identifying, for each of the multiple-frame groupings, a probability that the grouping includes a baseball hit based on how well the grouping matches the set of templates.

58. A method as recited in claim 57, wherein the extracting comprises, for each frame:

identifying an average waveform amplitude of the audio data in a first frequency band of the frame;

identifying an average waveform amplitude of the audio data in a second frequency band of the frame;

concatenating the identified average waveform amplitudes to generate a first energy feature of the set of features;

identifying an average waveform amplitude of the audio data in a third frequency band of the frame; and

using, as a second energy feature of the set of features, the average waveform amplitude of the third frequency band.

59. A method as recited in claim 58, further comprising:

generating a third energy feature for each frame by normalizing the first energy feature based on the first energy feature of the eighth frame of the multiple-frame grouping; and

generating a fourth energy feature for each frame by normalizing the second energy feature based on the second energy feature of the eighth frame of the multiple-frame grouping.

60. A method as recited in claim 44, wherein the combining comprises generating, for each of the first plurality of portions, a weighted sum of a probability that the portion includes excited speech and a probability that the portion includes a baseball hit.

61. A method as recited in claim 44, wherein the combining comprises adjusting the probability that a portion includes excited speech based on the probability that the portion includes a baseball hit.

62. A system comprising:

a feature extractor to extract a plurality of audio features from programming content;

an excited speech classification subsystem to identify, based on a sub-set of the audio features, a set of segments of the programming content and corresponding probabilities that the segments include excited speech;

a baseball hit detection subsystem to identify, based on another sub-set of the audio features, a set of frame groupings of the programming content and corresponding probabilities that the frame groupings include baseball hits; and

a probabilistic fusion subsystem to combine the probabilities that the segments include excited speech and the probabilities that the frame groupings include baseball hits, and to generate a probability that portions of the programming content are exciting based on the combination.

63. A system as recited in claim 62, wherein the excited speech classification subsystem is to identify the set of segments by:

identifying, based on a first set of the plurality of audio features, a plurality of windows of the programming content that include speech;

identifying, based on a second set of the plurality of audio features, which of the plurality of windows include excited speech.

64. A system as recited in claim 62, wherein the baseball hit detection subsystem is to identify the set of frame groupings by:

combining, for each of a plurality of multiple-frame groupings, a set of features from the programming content;

comparing the sets of features from the multiple-frame groupings to a set of templates; and

identifying, for each of the multiple-frame groupings, a probability that the grouping includes a baseball hit based on how well the grouping matches the set of templates.

65. A system comprising:

a receiving device to receive a sporting event;

a user interface to receive, from a user, an indication of a desired summary level for the sporting event; and

a processing subsystem to identify which portions of the sporting event to present to the user based at least in part on both the desired summary level and meta data corresponding to the sporting event, the meta data identifying a likelihood of each of a plurality of portions of the sporting event being exciting based at least in part on the presence of both excited speech and ball hits within the sporting event.

ABSTRACT

Audio/video programming content is made available to a receiver from a content provider, and meta data is made available to the receiver from a meta data provider. The meta data corresponds to the programming content, and identifies, for each of multiple portions of the programming content, an indicator of a likelihood that the portion is an exciting portion of the content. In one implementation, the meta data includes probabilities that segments of a baseball program are exciting, and is generated by analyzing the audio data of the baseball program for both excited speech and baseball hits. The meta data can then be used to generate a summary for the baseball program.

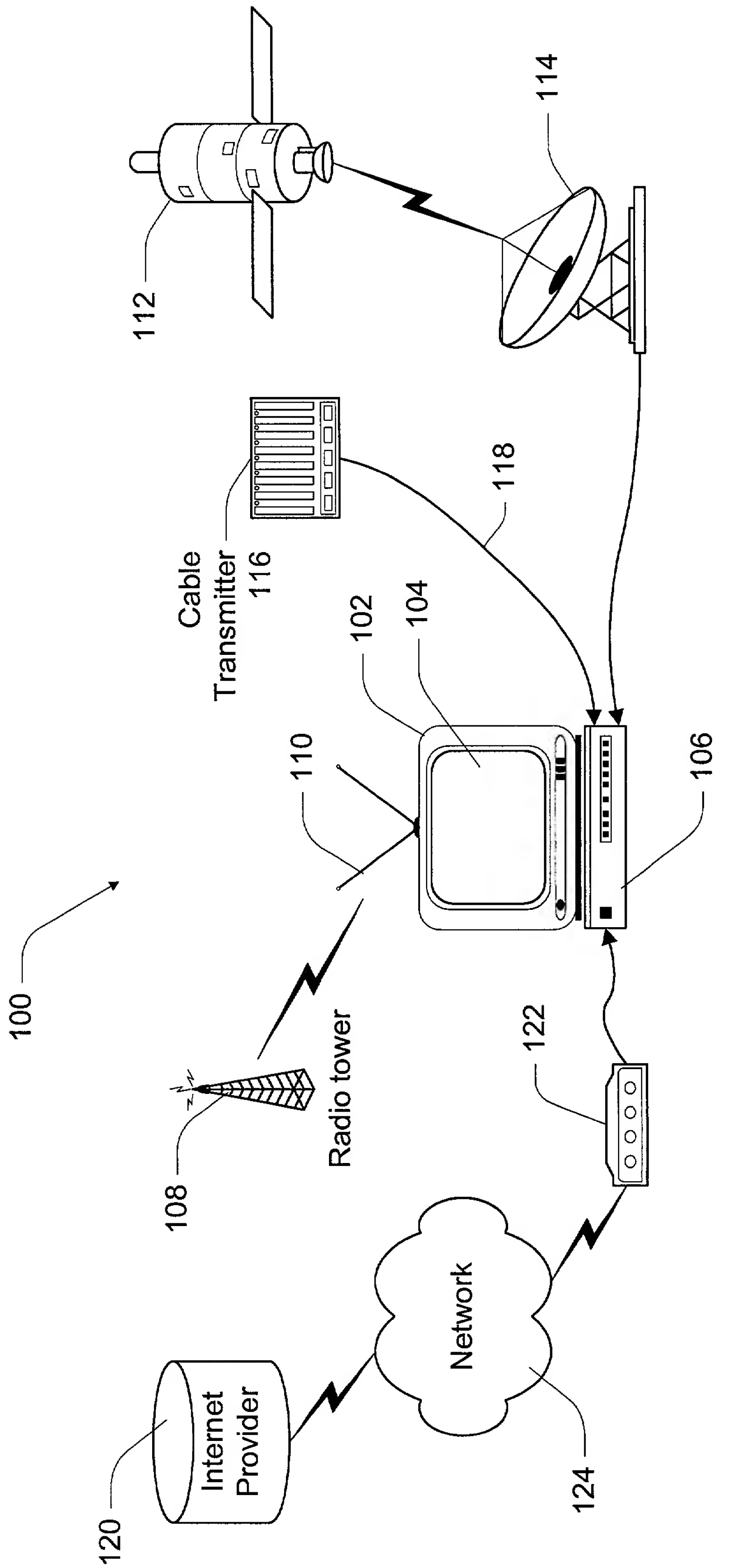
100
102
104
106
108
110
110
Radio tower
112
114
Cable Transmitter 116
118
Internet Provider
120
122
Network
124

Fig. 1

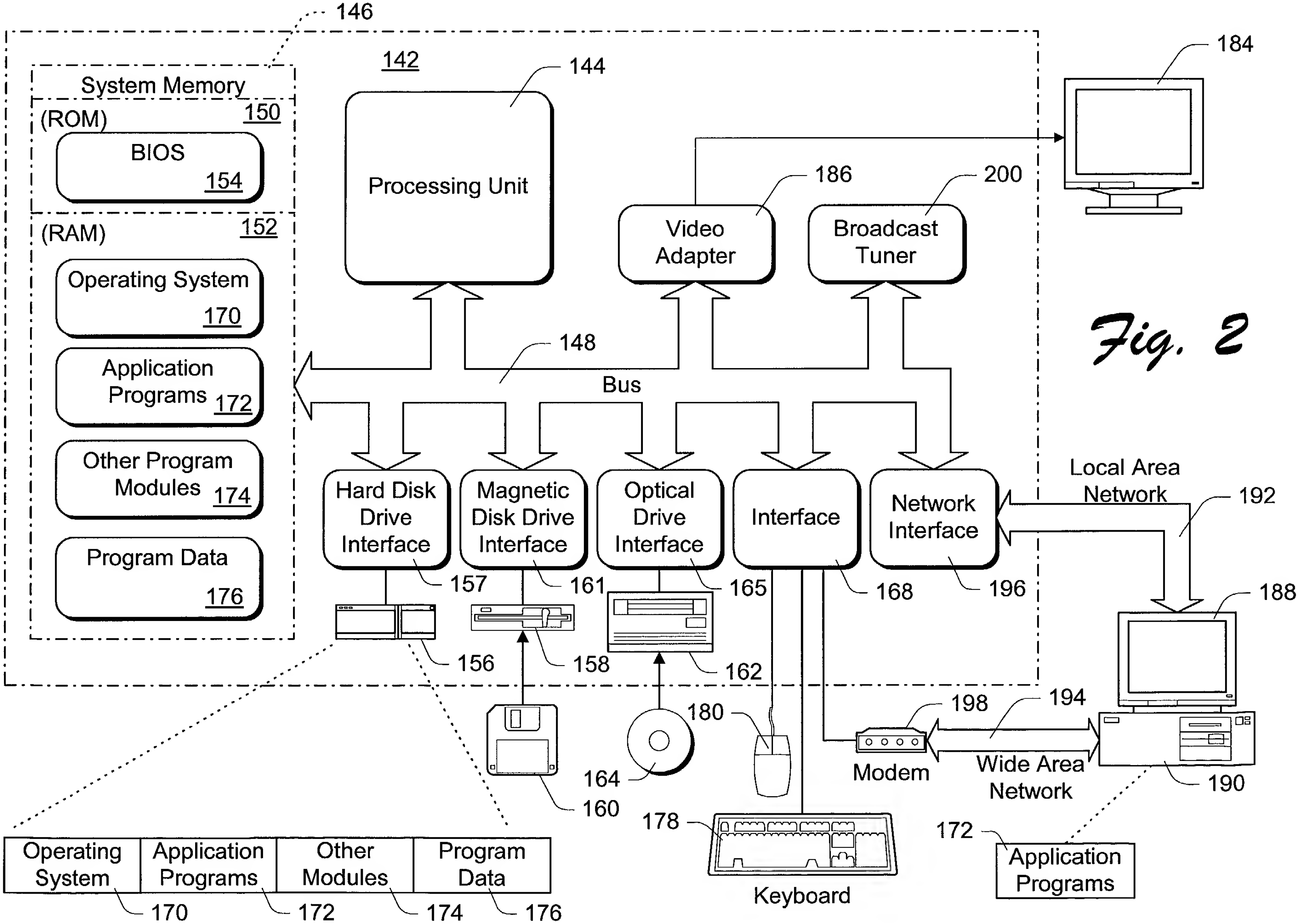
146
System Memory
(ROM)
150
BIOS
154
(RAM)
152
Operating System
170
Application Programs
172
Other Program Modules
174
Program Data
176
142
Processing Unit
144
Video Adapter
186
Broadcast Tuner
200
184
148
Bus
Hard Disk Drive Interface
157
Magnetic Disk Drive Interface
161
Optical Drive Interface
165
Interface
168
Network Interface
196
Local Area Network
192
188
156
158
162
180
198
194
Modem
Wide Area Network
190
164
160
178
Keyboard
172
Application Programs
Operating System
Application Programs
Other Modules
Program Data
170
172
174
176

Fig. 2

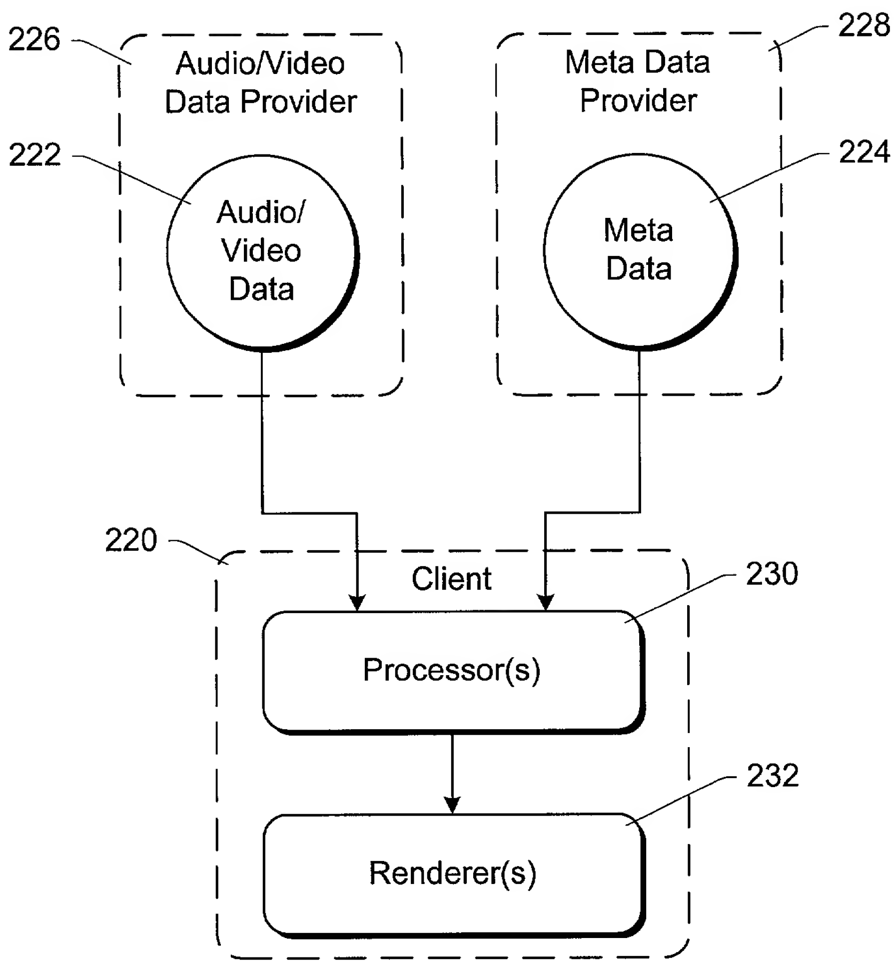
226
Audio/Video
Data Provider
222
Audio/
Video
Data
228
Meta Data
Provider
224
Meta
Data
220
Client
230
Processor(s)
232
Renderer(s)

Fig. 3

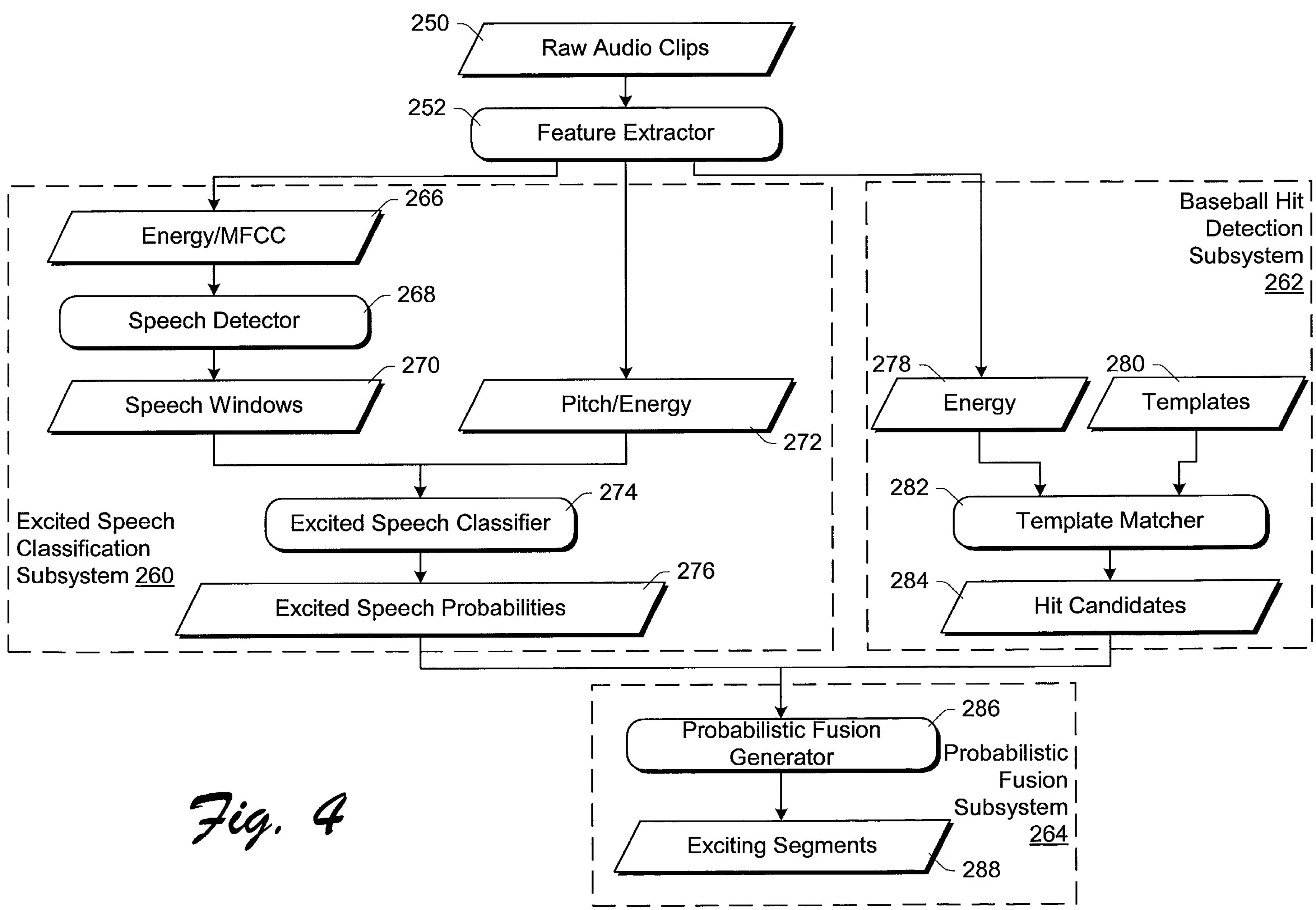
250
Raw Audio Clips
252
Feature Extractor
266
Energy/MFCC
268
Speech Detector
270
Speech Windows
Pitch/Energy
272
274
Excited Speech Classifier
Excited Speech Classification Subsystem 260
276
Excited Speech Probabilities
Baseball Hit Detection Subsystem 262
278
Energy
280
Templates
282
Template Matcher
284
Hit Candidates
286
Probabilistic Fusion Generator
Probabilistic Fusion Subsystem 264
Exciting Segments
288

Fig. 4

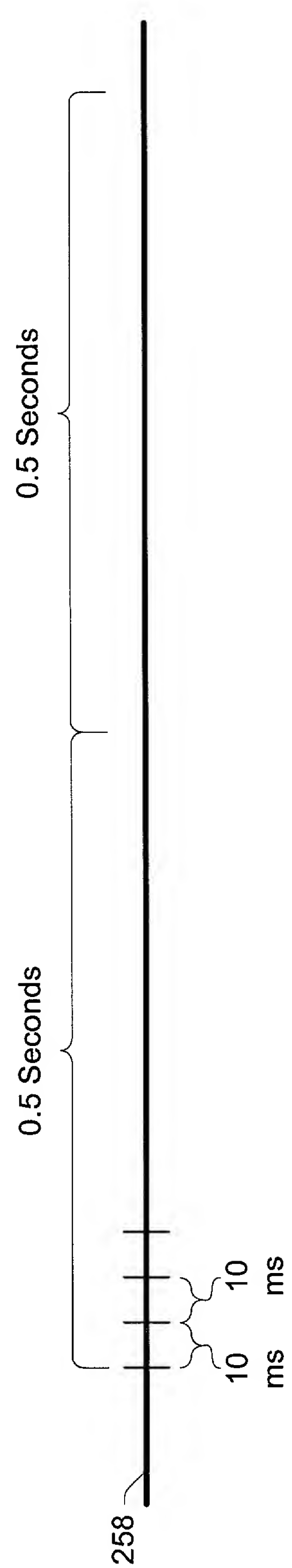

Fig. 5

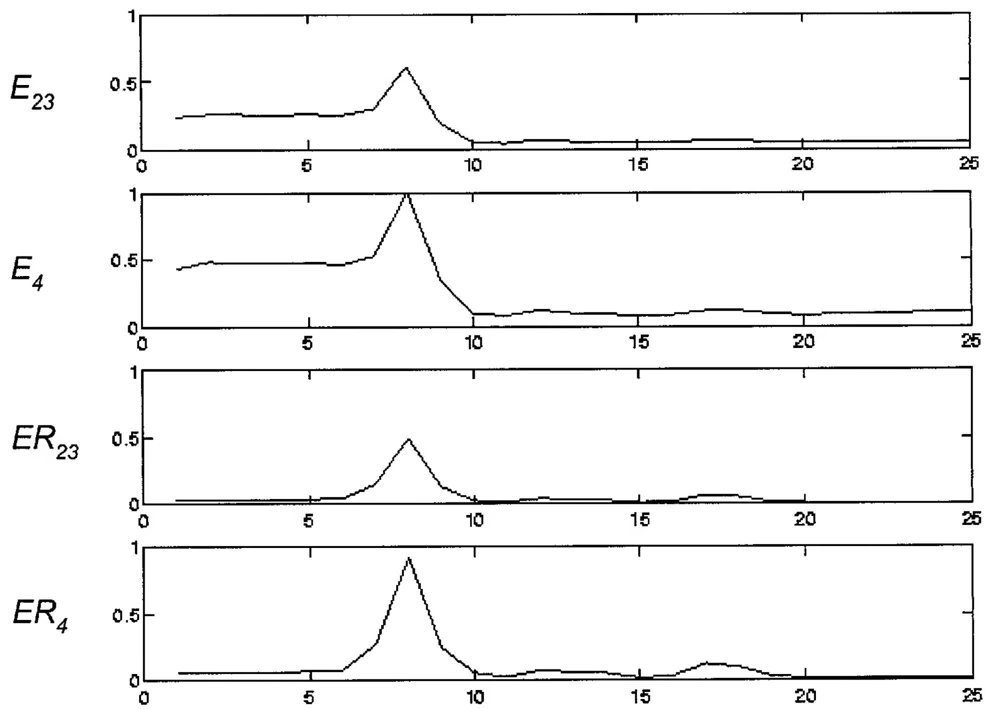

Fig. 6

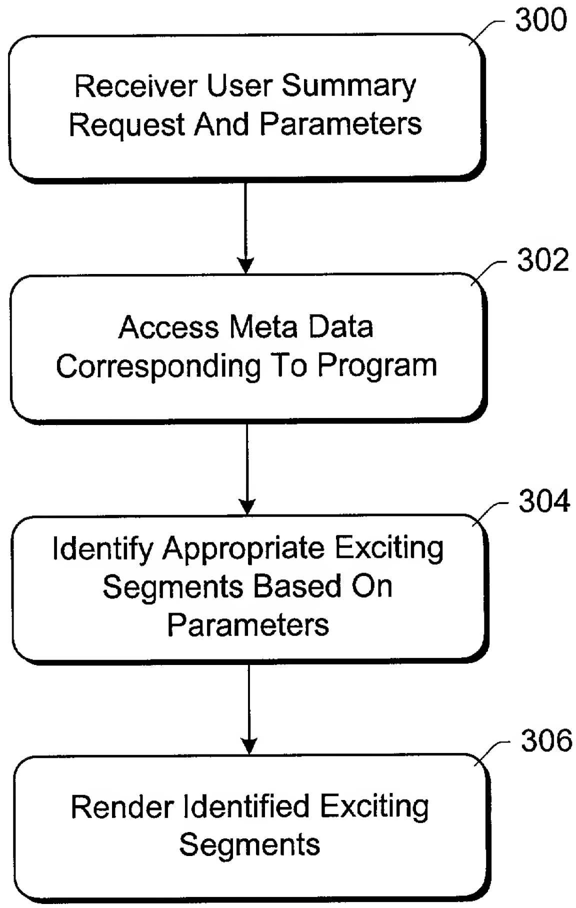
300
Receiver User Summary Request And Parameters
302
Access Meta Data Corresponding To Program
304
Identify Appropriate Exciting Segments Based On Parameters
306
Render Identified Exciting Segments

Fig. 7